REMARKS

Claims 1, 4-9 and 11-19 are pending in the application. Claims 1, 5 and 17 have been amended herein. Favorable reconsideration of the application is respectfully requested.

Initially, applicants want to thank the Examiner for taking the time to discuss the present application with the undersigned during a telephone interview on September 6, 2007. The Examiner was very helpful in facilitating applicants' understanding, and it was proposed that the applicants submit a formal response with amendments.

I. OBJECTION TO SPECIFICATION/REJECTION OF CLAIMS 1-19 UNDER 35 USC §112, 1ST ¶

The Examiner objects to the specification as not including support for the various claim recitations identified on page 2 of the Office Action. Relatedly, the Examiner rejects claims 1-19 under 35 USC §112, first paragraph, as not being supported by the specification. Applicants respectfully request withdrawal of the objection/rejection for at least the following reasons.

Applicants incorporate herein by reference the comments presented in their response after final rejection filed on August 7, 2007. Applicants address in detail each of the Examiner's concerns regarding support in the specification.

Regarding the particular claim amendments presented herein, applicants note that the present invention is described in the specification as including a ROM (e.g., 104) having cipher data (e.g., 107) stored therein. (See, Fig. 1). The cipher data is used for decrypting the encrypted intermediate code (e.g., 108) stored in RAM (e.g., 103). (Spec., p. 12, lns. 17-20).

The encrypted intermediate code includes address information indicating where the cipher data for decrypting the encrypted intermediate code is stored within the ROM. The address of the cipher data stored within the ROM is independent of where the encrypted intermediate code is stored in the RAM. For example, the specification describes how the encrypted intermediate code data 701 includes an encryption flag 702, a first address 703 of the cipher data on the ROM which is used for encrypting the intermediate code 701, a cipher data size 704, and the encrypted intermediate code 705. (Spec., p. 19, lns. 25-29). The interpreter execution program is configured to generate the control command strings from the intermediate code. (Spec., p. 10, lns. 17-23).

II. REJECTION OF CLAIMS 1, 12 AND 15-19 UNDER 35 USC §103(a)

Claims 1, 12 and 15-19 remain rejected under 35 USC §103(a) based on *Westheimer et al.* in view of *Buerkle et al.* Applicants respectfully request withdrawal of the rejection for at least the following reasons.

Claim 1, as amended, recites the aspect of the invention whereby cipher data used for decrypting the encrypted intermediate code is stored in ROM. Moreover, the encrypted intermediate code includes address information indicating where the cipher data for decrypting the encrypted intermediate code is stored within the ROM, the address of the cipher data being independent of where the encrypted intermediate code is stored in the RAM. The CPU executes the interpreter execution program for decrypting and generating the another command control string from the encrypted intermediate code by accessing the cipher data stored in the ROM at the address identified in the encrypted intermediate code. Claims 5 and 17 recited similar features.

Westheimer et al., whether alone or in combination with Buerkle et al., does not teach or suggest such features. More specifically, Westheimer et al. states:

> *The transformation enable circuit is initially activated by a memory boundary operation code which will normally be one of the first instructions in an encoded program.* (Westheimer, Col. 2, lns. 54-57; Emphasis Added).
>
> *Once activated, the transformation enable circuit monitors the flow of data and addresses between the central processing unit of the computer and the computer main memory. Data and addresses which fall within a segment of memory defined by the memory boundary operation code undergo transformation, while data and addresses which fall outside said segment remain unaffected.* (Col. 2, lns. 58-65; Emphasis Added).

Accordingly, Westheimer et al. teaches that an encoded program is stored in RAM 48. One of the first instructions incuded in the encoded program B is a memory boundary operation code that identifies boundary addresses of the RAM 48. These boundary addresses indicate that code stored within the boundary addresses are encoded, and code that falls outside the boundary addresses are non-encoded. (See, Col. 4, ln. 57 to Col. 5, ln. 12).

Westheimer et al. also describes ROMs 42 and 44 which would appear to include cipher data. However, the selection of the particular cipher data is *based on the address from where the encrypted data was obtained from the RAM 48*. (See, e.g., column 8, lines 20-68). Westheimer et al. does not teach or suggest that the encrypted data stored within the RAM 48 itself includes address information indicating where the cipher data is stored within the ROM independent of where the encrypted data is stored within the RAM 48.

More specifically, Fig. 2 of Westheimer et al. illustrates how the address of the encrypted data as input to the RAM 48 (e.g., A_{0-7} and A_{8-15}) determines the address of the cipher data in the ROMs 42 and 44 (A_{0-7} and A_{8-15} serve as addresses to the ROMs 42 and 44). Accordingly, to the extent the Examiner may consider Westheimer et al. to include address information indicating where the cipher data is stored in the ROM, the address is *not* independent of where the encrypted data is stored in the RAM as recited in claims 1, 5 and 17.

As pointed out in applicants' response filed on August 7, 2007 and discussed again below, neither Westheimer et al. nor Buerkle et al. teach or suggest intermediate code as recited in claims 1, 5 and 17. Rather, Westheimer et al. and Buerkle et al. relate to machine code. The presently claimed invention is advantageous in that machine code does not lend itself well to containing address information indicating where in the ROM the cipher data is stored. With the present invention, the intermediate code can be readily relocated in the RAM, whereas difficulty is experienced in attempting to relocate the machine code as in Westheimer et al.

Westheimer et al. and Buerkle et al. Do Not Teach/Suggest Intermediate Code

Applicants also previously argued that neither Westheimer et al. nor Buerkle et al. teach or suggest storing both intermediate code and encrypted intermediate code in RAM. Applicants pointed out that Westheimer relates to storing *operation* code (i.e., machine code instructions) and encrypted *operation* code in RAM, not *intermediate* code as claimed.

The Examiner responded by stating:

> *In response to applicants arguments against the references individually, one cannot show nonobviousness by attaching references individually where the rejections are based on combinations of references (citations omitted). Regarding the applicant's above argument, the examiner respectfully points out that claims 1 and 12 stand rejected in view of Westheimer and Buerkle.*
>
> *In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., Basic, Java, PASCAL, etc.) are not recited in the rejected claims.* (O.A., p. 10).

Regarding Westheimer, applicants referred to Basic, Java, PASCAL, etc., merely as examples of intermediate code, and to exemplify the distinction between intermediate code and machine code as in Westheimer. Westheimer does not teach or suggest the storage of intermediate code and encrypted intermediate code as recited in the claims.

Regarding Buerkle, this reference also fails to teach or suggest the storage of intermediate code and encrypted intermediate code in memory as recited in claims 1, 5 and 17. Thus, even if the references are viewed in combination as suggested by the Examiner, the references still do not teach or suggest the claimed invention. Neither reference teaches storing intermediate code and encrypted intermediate code in memory, and thus nor does the combination. The fact that the claims do not recite Basic, Java, PASCAL, etc. is not material to applicants' argument.

Applicants refer to the previously submitted articles, "A RISC Interpreter for the C Programming Language" by Davidson et al., "Techniques for Obtaining High Performance in Java Programs" by Kazi et al., and "Comparing Java Implementations for Linux" by Hirsch. These articles explain in more detail the differences between intermediate code and operation code. Namely, intermediate code is not dependent upon the CPU specification, whereas operation or machine code is dependent upon the CPU specification.

. No Interpretation Execution Program Stored in ROM

The Examiner admits that Westheimer does not disclose that the CPU operates using an "interpreter execution program" stored in ROM. However, the Examiner submits that Buerkle teaches such feature and thus makes up for the deficiencies of Westheimer. Applicants respectfully submit that Buerkle does not teach an "interpreter execution program" stored in ROM, and rather teaches away from such feature.

The Examiner asserts that Buerkle teaches that LSI processors utilize an "interpreter execution program" [microprogram] to allow a CPU to process intermediate code to allow a CPU to process intermediate code [macroinstructions] according to the opcodes of the instructions. The Examiner indicates that Buerkle teaches that prior art discloses LSIs as storing the "interpreter execution program" in a ROM (Buerkle, "Description of the Related Art"). (See, O.A., ¶ bridging p. 4-5).

Specifically, the Examiner indicates that it would have been obvious to one of ordinary skill in the art to recognize the need for an "interpreter execution program" stored in a ROM as recited in claims 1, 6 and 17 to allow a CPU to control the execution of an "intermediate code", and thus follow the LSI processor design teachings of Buerkle within the LSI processor system of Westheimer. The Examiner suggests this would have been obvious because one of ordinary skill in the art would have been motivated to practically implement the features known in the prior art to be included within LSI processor systems. (See, O.A., p. 5, lns. 4-10).

Applicants respectfully disagree with such conclusion. In fact, comparing the disclosure in the section "Description of the Related Art" in Buerkle with the disclosure in the "Description of the Related Art" in the present application, it is clear that the respective disclosures refer to the same problem with the conventional art.

Particularly, in the present application is taught that:

> *[i]n a conventional LSI, software used for <u>performing its operation is previously stored in a memory such as a ROM</u> or the like. When the LSI performs its operation, a <u>CPU reads software from the ROM</u>. In the above conventional LSI, <u>only software previously stored in the ROM can be executed, and it is not permitted to freely make a modification to software stored in the ROM</u>. Thus, <u>such an LSI cannot be used with various disc apparatuses available from a plurality of manufacturers</u>,... . Furthermore, modifying specifications or adding functions, a need for which may arise in the process of development of a disc apparatus, cannot be readily satisfied..." (Emphasis added; See, Spec., ¶ bridging p. 1-2).*

In the section "Description of the Related Art" in Buerkle, it is disclosed that:

> *Generally, a processor performs a series of operations upon digital information in an execution unit according to a stored program of instructions. These instructions are often termed "macroinstructions" in*

> *order to avoid confusion with the "microinstructions" contained in the control store of the processor. The microinstructions are often grouped into microinstruction routines in order to perform various macroinstructions. The grouped microinstruction routines constitute the microprogram of the processor. (Emphasis added; Col. 1, lns. 30-39).*
>
> *Prior art processors use microprogrammed instructions to control internal circuitry in response to macro-instructions. These processors use table look-up approaches to map a macroinstruction into the appropriate microinstruction routines/microprogram. ... Each micro-branch is accessed by an address formed directly from the operation-code (op-code) of the macroinstruction. In addition, the micro-branch entries are physically remote from the processor in firmware, a ROM, to permit flexibility during the manufacturing process. One disadvantage of this approach is that entries in the remote table cannot be directly manipulated by the processor. This eliminates flexibility to make changes caused by development and later engineering changes. A second disadvantage results from the micro-branching process which requires extra operating cycles when the branching operations are executing. These extra cycles result in considerable delay between the time a processor work sequence is initiated by a macroinstruction and the decode of the first useful microinstruction routine. (Emphasis added; Col. 1, lns. 40-65).*

In other words, Buerkle teaches that "microinstructions" are stored in ROM, and that "macroinstructions" are effective groupings of microinstruction routines that constitute the microprogram of the processor. Buerkle does not appear to teach that an interpreter execution program is stored in ROM to interpret intermediate code, and to decrypt and interpret encrypted intermediate code. Rather, macroinstructions are performed by mapping (using table look-up approaches) such macroinstructions into appropriate microinstruction routines/microprogram.

As mentioned above, this is similar to the "Related Art" as described in the present application. In both the present application and Buerkle, this storing of microinstructions in ROM is described as being disadvantageous.

Therefore, even if the microprogram of Buerkle were to be interpreted as "an interpreter execution program", Buerkle seems to actually teach away from such microprogram being stored in ROM (see, particularly, Col. 1, lns. 51-65).

Furthermore, applicants respectfully submit that the microprogram is not the "interpreter execution program" of the present claimed invention, contrary to the Examiner's assertions. Rather, the microprogram represents the command control string to be executed by the control section, so that the microprogram more clearly resembles the intermediate code of the present claims.

The "program that interprets" the microprogram includes the table look-up approaches that map the macroinstruction into the appropriate microinstruction routines/microprogram.

In summary, a person of ordinary skill in the art studying Buerkle would not be motivated to store an interpreter executing program in ROM, because (1), even if the "microprogram" of Buerkle were to be interpreted as an "interpreter executing program", Buerkle teaches away from storing the microprogram in ROM due to the disadvantages as highlighted above, (2) the "microprogram" of Buerkle is *not* an "interpreter executing program" (since it does not interpret the intermediate code; but is actually more representative of the intermediate code itself), and (3) Buerkle does not actually teach or suggest a program used for interpreting intermediate code, wherein the interpreter execution program is stored in ROM.

Therefore, Buerkle does not teach the feature "a ROM for storing an *interpreter execution program* that is configured to interpret the intermediate code, and to decrypt and interpret the encrypted intermediate code, as recited in claims 1, 6 and 17.

For at least the reasons discussed above, applicants respectfully submit that Westheimer and Buerkle, taken alone or in combination, do not teach or suggest the present invention as claimed. Applicants respectfully request that the rejection be withdrawn.

III. REJECTION OF CLAIMS 4-14 UNDER 35 USC §103(a)

Claims 4-14 stand rejected under 35 USC §103(a) based on *Westheimer et al.* in view of *Buerkle et al.*, and further in view of *Hagiwara et al.* Applicants respectfully request withdrawal of this rejection for at least the following reasons.

Regarding independent claim 5, this claim recites features similar to claim 1 as discussed above. Accordingly, claim 5 may be distinguished over the teachings of *Westheimer et al.* and *Buerkle et al.* for at least the same reasons expressed above. Moreover, *Hagiwara et al.* does not make up for the above-discussed deficiencies in *Westheimer et al.* and *Buerkle et al.*

The remaining claims represent dependent claims that depend from either claim 1 or claim 5 discussed above, and may be distinguished for at least the same reasons.

IV. REJECTION OF CLAIMS 1-19 UNDER 35 USC §103(a)

Claims 1-19 now stand rejected under 35 USC §103(a) based on *CN2045628U* in view of *CN1245926A*. Applicants respectfully request withdrawal of this rejection for at least the following reasons.

In rejecting the claims, the Examiner points to the rationale expressed in the Office Action dated July 9, 2004 in relation to the corresponding Chinese application.

The Chinese Examiner asserts that the difference between claim 1 and *CN2045628U* is the content stored in the RAM and ROM. The Chinese Examiner asserts that the content stored in storage can be voluntarily appointed based on particular application, which does not need creative effort for the person skilled in the art. The Chinese Examiner further asserts that the technological problem to be solved by claim 1 and *CN2045628U* is to integrate RAM, ROM and CPU into one chip.

Applicants note, on the other hand, even though *CN2045628U* does disclose that RAM, ROM and CPU are integrated into one chip, the RAM does not store an intermediate code and the ROM does not store an interpreter execution program which is capable of interpreting the intermediate code as claimed. *CN1245926A* does not make up for such deficiencies.

Accordingly, applicants respectfully request withdrawal of the rejection.

V. CONCLUSION

Accordingly, all claims 1, 4-9 and 11-19 are believed to be allowable and the application is believed to be in condition for allowance. A prompt action to such end is earnestly solicited.

Should the Examiner feel that a telephone interview would be helpful to facilitate favorable prosecution of the above-identified application, the Examiner is invited to contact the undersigned at the telephone number provided below.

Should a petition for an extension of time be necessary for the timely reply to the outstanding Office Action (or if such a petition has been made and an additional extension is necessary), petition is hereby made and the Commissioner is authorized to charge any fees (including additional claim fees) to Deposit Account No. 18-0988.

Respectfully submitted,

RENNER, OTTO, BOISSELLE & SKLAR, LLP

/Mark D. Saralino/
Mark D. Saralino
Registration No. 34,243

DATE: October 9, 2007

The Keith Building
1621 Euclid Avenue
Nineteenth Floor
Cleveland, Ohio 44115
(216) 621-1113

B:\GEN\YAMA\Yamap797\yamap797responseaccompanyingRCE6.wpd